Rapport projet C++

Demuth Axel

December 2023

1 Introduction

Dans le cadre du projet visant à étudier le comportement d'un système de refroidissement d'un micro processeur,j'ai codé un programme permettant de simuler numériquement le comportement d'une partie du système.Le système de refroidissement est composé de plusieurs ailette qui sont des éléments très conducteur de chaleur permettant de refroidir ainsi le composant électronique.

Pour cela nous étudions le problème sous deux forme , un état stationnaire ou nous regardons la température de l'ailette à partir d'un certain temps t, on suppose qu'à partir de ce moment t l'ailette rentre dans l'état stationnaire. Nous regarderons alors la température tout au long de l'ailette. L'autre méthode est de regardons l'évolution de la température en fonction du temps; on appellera ce cas : le cas instationnaire.

On regardera donc d'abord le cas stationnaire puis dans un second temps le cas instationnaire, dans les deux cas nous étudierons la discrétisation du problème et la résolution numérique effectué.Pour cela nous allons regarder l'équation de la chaleur suivante :

$$\rho C_p \frac{\partial T}{\partial t} - k\frac{\partial^2 T}{\partial x^2} + \frac{h_c p}{S}(T - T_e) = 0$$

avec les conditions au bord suivantes :

$$-k\frac{\partial T}{\partial x} = \phi_p$$

$$-k\frac{\partial T}{\partial x} = 0$$

L'ensemble des paramètres seront les suivants :
Lx,Ly,Lz la longueur,hauteur,largeur de l'ailette,
ϕ_p le flux de chaleur , Te la température ambiante,
ρ la densité, C_p la chaleur à pression constante,
k la conductivité thermique h_c le coefficicent de transfert de chaleur surfacique;
S = LyLz la surface l'aire transversale et p = 2*(Ly+Lz) le périmètre de la section transversale.

2 Cas stationnaire

Nous commençons par traiter le cas stationnaire, donc lorsque t tend vers l'infini. Pour résoudre l'équation de la chaleur nous utilisons la méthode des différences finies. On découpe le segment [0, Lx] en M intervalles de longueur h = Lx/M, avec xi = i*h , i = 0,..,M la solution numérique.On note Ti la température au point xi.
En utilisant un développement de taylor on discrétise la dérivée seconde avec le schéma suivant

$$\frac{\partial^2 T}{\partial x^2} \approx \frac{T_{i-1} - 2T_i + T_{i+1}}{h^2}$$

Ce qui nous donne

$$-k\frac{T_{i-1} - 2T_i + T_{i+1}}{h^2} + \frac{h_c p}{S}(Ti - Te) = 0$$

avec en condition au bord en 0 puis en Lx

$$-k\frac{\partial T}{\partial x} \approx -k\frac{T_1 - T_0}{h} = \phi_p$$

$$-k\frac{\partial T}{\partial x} \approx -k\frac{T_{M-1} - T_M}{h} = 0$$

On peut donc résumer ce problème en une équation de type AX = F, A une matrice Tridiagonal (M * M),F et X des vecteurs de taille M, X donnant la température en chaque point xi .
Pour résoudre cette équation on utilisera une décomposition LU.L'avantage de cette méthode est qu'il est très simple de trouver la décomposition LU des matrices Tridiagonal .On peut donc trouver L et U très facilement.Puis avec un algorithme de monter,descente nous trouvons X.

Pour Cela j'ai fais un programme en deux fichier, un fichier tridiag.hpp comprenant un constructeur de matrice tridiagonal à partir des trois diagonales et la taille de la matrice,un méthode pour afficher la matrice au besoin, utile pour debug le code,puis une méthode résolution Lu qui prend en entrée le vecteur F, cet méthode effectue la décomposition LU de la matrice tridiag puis effectue la montée descente pour calculer X.

En plus de ce fichier tridiag.hpp,j'ai un autre fichier, stationnaire.cxx qui permet d'effectuer la résolution Lu et d'écrire le vecteur X dans un fichier csv .Pour faire fonctionner ce programme il faut lui donner un fichier texte en argument,il faut que le ficher est la même forme que le fichier suivant :

```
1 Lx 0.040 Ly 0.004 Lz 0.05
2 M 5000
3 Phi 12500
4 hc 200
5 rho 2700
6 k 164
7 Cp 940
8 te 20
9 TFinal 300
10 N 600
11 Mx 50 My 10 Mz 30
```

Figure 1: exemple de fichier simu.txt

Ce fichier sera lu par un programme c++ dans notre cas stationnaire.cxx .Après avoir importer le fichier, le programme va calculer les diagonales de A, qui sont données par les trois vecteurs suivants : a,b,c ; b étant la diagonal de la Matrice A , a la diagonal en dessous et c la diagonal au dessus.On a :

$$a_i = -\frac{k}{h^2};\ c_i = -\frac{k}{h^2};\ b_i = \frac{2k}{h^2} + \frac{h_c p}{Te};\ \forall i \in [1, M-1]$$

$$a_{M-1} = -\frac{k}{h};\ c_0 = -\frac{k}{h};\ b_0 = b_M = \frac{k}{h}$$

et le vecteur second membre F comme ceci :

$$F_0 = \phi;\ F_M = 0;\ F_i = \frac{h_c p}{S} Te\ \forall i \in [1, M-1]$$

En utilisant la bibliothèque matplotlib de python nous pouvons afficher la courbe de l'évolution de la température en fonction de la positions du point xi sur l'ailette;vous avez ce code a disposition dans le fichier plt.py ou il suffit de remplacer le M avec votre nombre de point que vous avez utiliser dans votre fichier simu.txt.Les paramètres proposés dans le fichier en exemple nous donne la courbe suivante :

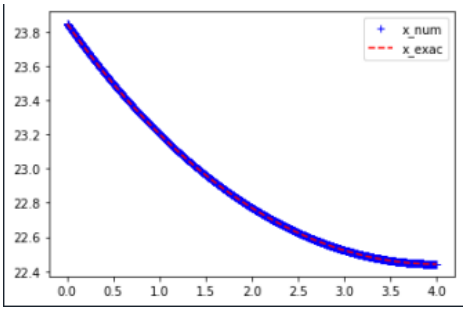
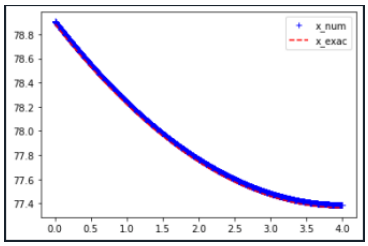
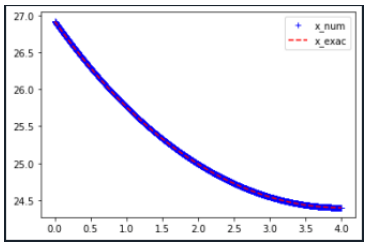


Figure 2: Courbe cas stationnaires par simu.txt

avec xnum la solution que l'on a calculer, et xexac la température théorique en chaque point,nous pouvons observer que les deux courbes se superposent bien donc on peut supposer que notre modèle est bon.Nous pouvons expérimenter aussi le comportement de la température en changeant quelques paramètres:
Par exemple en éteignant le ventilateur du processeur ce qui revient à hc = 10;
ou en augmenter le flux de chaleur



(a) Courbe avec ventilateur éteint



(b) Courbe avec phi = 22500

On voit sur le premier graphe qu'avec le ventilateur éteint la température explose , ici elle fait un peu plus que triplé.Puis sur le deuxième avec une augmentation du flux de chaleur qui passe de 12500 a 22500 et un ventilateur allumé la température n'augmente que de 3-4 degré.

3 Cas Instationnaire

Dans cette partie nous allons rajouter une dimension au problème, la notion de temps.Nous voulons pouvoir étudier l'évolution au cours du temps de la température en un point.Nous devons donc changer un peu notre équations, Notons Tfinal le temps jusqu'au quel nous mesurons la température, N le nombre de pas de temps , Δt = N/Tfinal et $t_n = n\Delta t \ \forall n \in [0, N]$ l'échantillonage du temps.On note T_i^n la température au temps tn a la position xi.

En discrétisant avec une méthode de Euler nous avons :

$$\frac{\partial T}{\partial t}(xi, tn) \approx \frac{T_i^{n+1} - T_i^n}{\Delta t}$$

nous obtenons ainsi les équations du problème instationnaire sur tout les noeuds de temps suivantes :

$$\rho Cp \frac{Ti^{n+1} - T_i^n}{\Delta t} - k \frac{T_{i-1}^{n+1} - 2T_i^{n+1} + T_{i+1}^{n+1}}{h^2} + \frac{h_c p}{S}(T_i^{n+1} - Te) = 0$$

avec en condition au bord en 0 puis en Lx

$$-k\frac{\partial T}{\partial x} \approx -k\frac{T_1^{n+1} - T_0^{n+1}}{h} = \phi_p$$

$$-k\frac{\partial T}{\partial x} \approx -k\frac{T_{M-1}^{n+1} - T_M^{n+1}}{h} = 0$$

On a donc une équation assez proche de celle du mode stationnaire, on doit réajuster le vecteur b et le vecteur second membre F et nous pourrons réutiliser la méthode de résolution LU. On aura le vecteur a et c qui resteront les mêmes, b et F qui seront inchangé au bord mais réajuster sur le reste comme ceci :

$$b_i = 2 * \frac{k}{h^2} + \frac{H_c p}{S} + \frac{\rho C_p}{\delta t}$$

$$F_i = T_i^n \frac{\rho C_p}{\delta t} + \frac{H_c * p}{S} Te$$

On aura alors A $T_i^1 = F_i$, Il suffira alors pour calculer T_i^{n+1} d'itérer sur n la formule suivante :

$$AT_i^{n+1} = T_i^n \frac{\rho C_p}{\delta t} + \frac{H_c * p}{S} Te$$

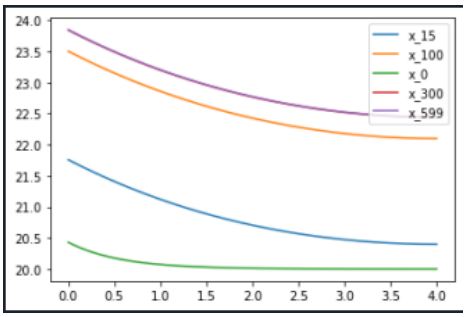
Par rapport au cas stationnaire il faut donc actualiser le second membre à chaque itération de n. On peut alors a l'aide du programme instationnaire.cxx simuler ce cas.Pour cela nous devons lui donner un fichier simu.txt de la même forme du cas stationnaire en rajoutant a la fin un entier n qui est le nombre de tn dont nous voulons les données, et un nombre n de temps tn qui sont les moments où nous récupéreront les températures en tout point. en exemple cela

revient à ajouter :

```
n 5
t 0 15 100 300 599
```

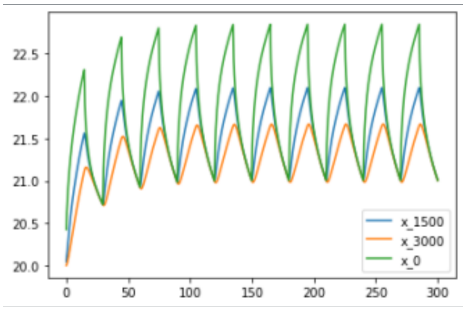
Enter Caption

Le t maximal que nous pouvons donner étant N-1.Avec les paramètres données dans les exemples de fichier simu.txt nous permettent alors de tracer les courbes suivantes:



courbe de la température à certain t

Nous observons alors la croissance au cours du temps de la température en tout points, et en observant la courbe a tn = 599, on remarque que les températures sont proches de celle du cas stationnaire, nous pouvons alors supposer que notre programme fonctionne bien.
On peut aussi essayer de voir comment se comporte la température de l'ailette quand l'on éteint et rallume la source de chaleur, on se pose comme base que l'on passe $\phi = 0$ et $\phi = 12500$ la valeur donnée dans le fichier simu.txt, on a alors la courbe de température en certain point par rapport au temps du suivante :



température avec flux de chaleur variant en certain point

4 Visualisation 3D

On a donc pu observer le comportement de la température dans le cas 1d , nous pouvons maintenant essayer de voir la progression de la chaleur sur une ailette 3D.Pour cela nous allons faire un nouveau maillage avec les paramètres, Mx , My, Mz comme dans l'exemple de fichier simu.Txt vu plus tôt.Les trois paramètres servent respectivement à diviser la longueur la hauteur et la largeur de l'ailette en Mx,My,Mz point. Nous supposons que la température ne change que en fonction de la distance en x par rapport à la source de chaleur.C'est à dire que la position sur y et z n'influe pas sur la température.Notons les points de ce maillage $P_{i,j,k}$, la hauteur et la largeur n'influent pas sur la température nous regardons les points uniquement par rapport à x, nous les nommerons x_{ijk} ou $x_{ijk} = x_{i00} \forall j,k \in [0, My] \times [0, Mz]$. Pour avoir des valeurs de la température nous allons regarder la Température dans les points x_k et x_{k+1} plus proches de nos points $P_{i,j,k}$ et nous noterons la température en ce point P : $\hat{T}$.
Pour cela nous allons interpoler ce point avec les droites suivantes :

$$T_k = ax_k + b \quad T_{k+1} = ax_{k+1}$$

a et b étant inconnu nous les trouvons comme ceci :

$$a = \frac{T_{k+1} - T_k}{x_{k+1} - x_l} \quad b = ax_k - T_k$$

ce qui donne à la fin :

$$\hat{T}_i = ax_{i00} + b$$

Nous stockerons tous les $\hat{T}_i$ dans un fichier sous format vtk, l'écriture de ces fichiers .Vtk se feront avec les programmes visualisation.cxx et visualisationt.cxx pour le cas stationnaire et pour le cas instationnaire, dans les deux cas il faudra donner en argument le fichier simu.txt à la compilation.

Ce programme tourne principalement autour de la classe visualisation ou l'on doit donner Lx et Mx en argument pour fabriquer le vecteur x_{i00} , une fonction pour afficher pour debuguer et une fonction calculTi qui a besoin de la discrétisation de l'espace de Lx , de l'entier M qui est le nombre de point, la température initial Te et la température au moment voulu , donc ici le résultat du cas stationnaire pour notre premier cas. Cette méthode va chercher les x_k et x_{k+1} puis le a et b , pour fabriquer et retourner $\hat{T}$. Le main va ensuite récupérer les données de simu.txt pour faire tourner ce programme et nous donner le fichier vtk, avec les données montré plus haut nous obtenons ceci comme résultat:

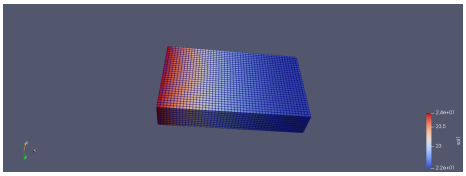
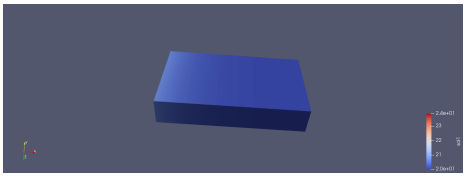
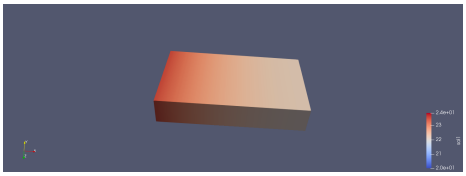


Figure 7: Réprésentation 3D du cas stationnaire

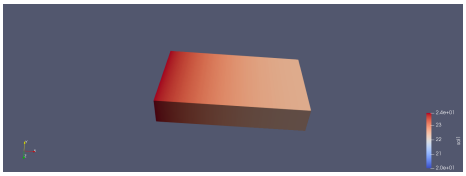
Dans la même logique en stockant dans un fichier vtk les données pour chaque T_i^n pour le modèle instationnaire nous pouvons obtenier une observation du comportement de la chaleur sur l'ailette dans le temps:



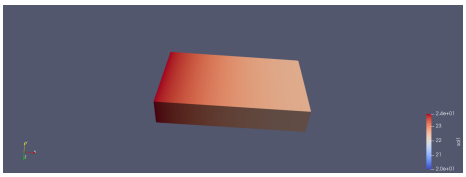
Représentation 3D du cas instationnaire a t0



Représentation 3D du cas instationnaire a t100

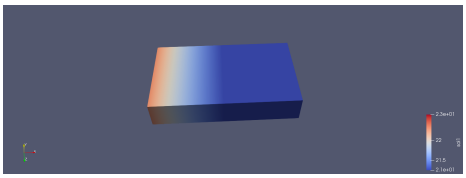


Représentation 3D du cas instationnaire a t300

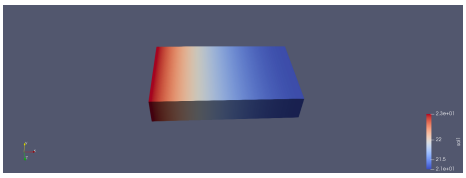


Représentation 3D du cas instationnaire a t599

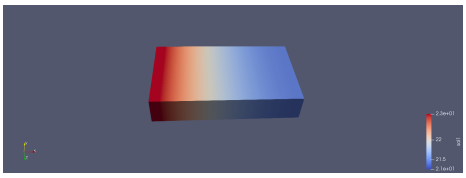
Nous pouvons alors observer la progression croissante de la température depuis la source de chaleur vers l'autre bout de l'ailette.
En plus de cela nous pouvons aussi simuler le cas ou nous faison variant le flux de chaleur , dans ce cas nous remarquons que le chaleur augmente de plus en plus au cours des cycles , je prends comme moment du cycle le moment ou la température atteint son maximal avant de retomber:



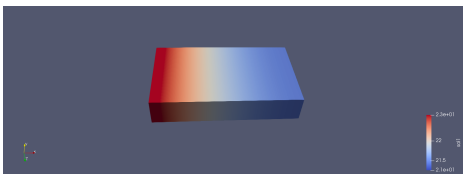
Représentation cas instationnaire phi variant premier cycle



Représentation cas instationnaire phi variant deuxième cycle



Représentation cas instationnaire phi variant cinquième cycle



Représentation cas instationnaire phi variant dixième cycle

Ainsi avec l'ensemble des simulations vu précedement, nous avons pu voir le comportement de la température sur une ailette du micro-processeur ,Ce qui resterait à faire désormais serait de pouvoir observer le comportement de la température avec plusieurs ailette pour réellement étudier à quel point la dissipation de chaleur serait efficace, on peut supposer qu'avec plusieurs La surface d'échange d'air serait aggrandi et le refroidissement serait encore plus efficace que comme vu avec une seule.